OCTASOFT
BASIC V7.0



C PLUS/4

C 16 /60 K/

CARTRIDGE

Előszó

Ez a leírás az új, és megváltoztatott utasítások, hibaüzenetek, rendszerváltozók és tokenek részletes szintaxisát tartalmazza, példákkal illusztrálva.

Az OCTASOFT BASIC V7.0 bővítő szoftver azoknak készült, akik a beépített BASIC nyelv adta lehetőségeket már nem találják kielégítőnek, és még nem járatosak az assembler programozásban. Az új BASIC utasítások segítségével több, magasabb szintű feladat oldható meg.

Néhány jellemző adat:

- A meglévő BASIC V3.5 utasításokon kívül további 100 új BASIC utasítás.
- A programsorok cimkével láthatók el.
- C-64 kompatibilis sprite-ok, beépített sprite-szerkesztő.
- Közvetlen hangkezelő utasítások.
- Lemezkezelő utasítások.
- Definiálható kitöltési minták a grafikus utasításokhoz.
- Programozható karakterkészletek.
- Tokenkompatibilitás a C-128-cal (A C-128-on írt BASIC programok jelentősebb változtatás nélkül futtathatók a C-PLUS/4-en).
- A program cartridge-ban kerül forgalomba, a felhasználható BASIC tárterület nem csökken.

Szerzők: ifj. Gulyás László
Szöllősi György

NOVOTRADE RT.
OCTASOFT STÚDIÓ

Budapest, 1988.

OCTASOFT BASIC V7.0 utasítások;

APPEND utasítás

Szintaxis: APPEND #logikai fileszám, filenév[,D meghajtó][,U egységszám]

Értelmezés: Megnyit egy meglévő szekvenciális file-t továbbírásra és pozícionálja az írófejet a file végére.

Logikai fileszám: Egy egész szám az 1-255 intervallumban. 128-nál kisebb érték megadásánál a PRINT# paranccsal kiírt adatok után egy kocsivissza-kód is kiíródik. 127-nél kisebb érték megadása esetén egy sorvége-kód kerül kiírásra.

Filenév: Egy kifejezés idézőjelek között vagy változóban, maximum 16 karakter hosszban.

Meghajtó: 0 vagy 1, attól függően melyik meghajtót kivánjuk használni. Alapesetben 0.

Egységszám: Egy egész szám a 4-15 intervallumban. Alapesetben 8.

Megjegyzés: A filenév, meghajtó és egységszám változóban is megadható, ilyenkor a változót zárójelbe kell tenni.

Példa 1:

```
APPEND #2,"SZöVEG"
```

Megnyitja a 8-as lemezegység 0-ás meghajtójában lévő lemezen található SZöVEG nevű file-t 2-es fileszámmal és az írófejet a file végére pozícionálja.

Példa 2:

```
APPEND #3,(F$),D(VAL(M$)) ON U9
```

A filenevet és a meghajtó számát változóban adtuk meg.

BEGIN és BEND utasítások

Szintaxis:

```
IF ... THEN BEGIN
.
. utasítások
.
BEND:ELSE BEGIN
.
. utasítások
.
BEND
```

Értelmezés: A BEGIN és BEND utasításpár közrefogja a BASIC utasítások csoportját, melyeket ezáltal több sorba írhatunk.

Megjegyzés: BEGIN utasítás csak THEN és ELSE utasítások után állhat. BEGIN utasítás után a BEND utasítás használata kötelező!

Példa 1:

```
100 IF X=1 THEN BEGIN:A=5
110 B=6:C=7
120 PRINT A+B+C:BEND:PRINT"AHA!"
130 ...
```

AHA! kiírása csak az x=1 logikai értéknél lesz, különben a 130-as programsorra kerül a vezérlés.

Példa 2:

```
10 INPUT"SZÁM=";Z
20 IF Z>0 THEN BEGIN
```

```
30 PRINT"A SZÁM POZITÍV"
40 BEND:ELSE GOTO 60
50 END
60 PRINT"A SZÁM NEM POZITÍV"
70 GOTO 10
```

Pozitív szám megadása esetén a program az 50-es sorban befejeződik, különben a 10-es sorban folytatódik.

BLOAD utasítás

Szintaxis: BLOAD filenév[,D meghajtó][,U egységszám][,P cím]

Értelmezés: Adatfile betöltése a memóriába tetszőleges címtől kezdődően.

Filenév: Egy kifejezés idézőjelek között vagy változóban, maximum 16 karakter hosszban.

Meghajtó: 0 vagy 1, attól függően melyik meghajtót kivánjuk használni. Alapesetben 0.

Egységszám: Egy egész szám a 4-15 intervallumban. Alapesetben 8.

Cím: Egy egész szám a 0-65535 intervallumban, a betöltés kezdőcíme. Amennyiben nincs megadva, úgy a kimentési címtől kezdődően történik a betöltés.

Megjegyzés: A filenév, meghajtó, egységszám és cím változóban is megadható, ilyenkor a változót zárójelbe kell tenni.

Példa 1: BLOAD"KÉP",P2048

A KÉP adatfile betöltése a 8-as lemezegység 0-ás meghajtójából, 2048-as címtől kezdődően.

Példa 2: BLOAD(F$),D(M),U(E),P(C)

Ebben az esetben minden paramétert változóban adtunk meg.

BOOT utasítás

Szintaxis: BOOT filenév[,D meghajtó][,U egységszám]

Értelmezés: Gépikódú programfile betöltése és elindítása a betöltési címtől kezdődően.

Filenév: Egy kifejezés idézőjelek között vagy változóban, maximum 16 karakter hosszban.

Meghajtó: 0 vagy 1, attól függően melyik meghajtót kivánjuk használni. Alapesetben 0.

Egységszám: Egy egész szám a 4-15 intervallumban. Alapesetben 8.

Megjegyzés: A filenév, meghajtó és egységszám változóban is megadható, ilyenkor a változót zárójelbe kell tenni.

Példa 1: BOOT "SEGÍTSÉG"

Betölti a 8-as lemezegység 0-ás meghajtójában lévő lemezen

található SEGíTSÉG nevü file-t és elindítja a kezdőcímétől kezdődöen.

BSAVE utasítás

Szintaxis: BSAVE filenév[,D meghajtó][,U egységszám],P cím1 TO P cím2

Értelmezés: A memória tetszöleges részének kimentése az adott file-ba.

Filenév: Egy kifejezés idézöjelek között vagy változóban, maximum 16 karakter hosszban.

Meghajtó: 0 vagy 1, attól függően melyik meghajtót kívánjuk használni. Alapesetben 0.

Egységszám: Egy egész szám a 4-15 intervallumban. Alapesetben 8.

Cím1, Cím2: Két egész szám a 0-65535 intervallumban, a kimentendő memóriarész kezdőcíme és végcíme + 1.

Megjegyzés: A filenév, meghajtó, egységszám és cím változóban is megadható, ilyenkor a változót zárójelbe kell tenni.

Példa 1: BSAVE"KÉP",P2048 TO P4096

A képernyő tartalmának kimentése a KÉP file-ba.

Példa 2: BSAVE(F$),D(M),U(E),P(C1) TO P(C2)

Ebben az esetben minden paramétert változóban adtunk meg.

BUMP függvény

Szintaxis: v=BUMP(n)

Értelmezés: A legutolsó sprite-kirajzolásnál (PROJECT) létrejött ütközések lekérdezése.

n: Értéke 1 vagy 2 lehet.

1: Sprite-sprite ütközések lekérdezése.
2: Sprite-háttér ütközések lekérdezése.

Megjegyzés A BUMP függvény értéke a 0-255 (%00000000-%11111111) intervallumba esik, az egyes ütközéseket a felemelt bitek jelzik.

Példa:

```
100 IF BUMP(1) AND 8 THEN 300
110 IF BUMP(2) AND 4 THEN 200
...
200 PRINT "HÁTTÉRREL ÜTKÖZÖTT A 3. SPRITE."
...
300 PRINT "SPRITE-TAL ÜTKÖZÖTT A 4. SPRITE!"
```

BVERIFY utasítás

Szintaxis: BVERIFY filenév[,D meghajtó][,U egységszám][,P cím]

Értelmezés: Ellenőrzi, hogy az elmentett file tartalma azonos-e a memória tetszőleges címtől kezdődő részével.

Filenév: Egy kifejezés idézőjelek között vagy változóban, maximum 16 karakter hosszban.

Meghajtó: 0 vagy 1, attól függően melyik meghajtót kívánjuk használni. Alapesetben 0.

Egységszám: Egy egész szám a 4-15 intervallumban. Alapesetben 8.

Cím: Egy egész szám a 0-65535 intervallumban, az ellenőrzés kezdőcíme. Amennyiben nincs megadva, úgy az ellenőrzés a kimentési címtől kezdődően történik.

Megjegyzés: A filenév, meghajtó, egységszám és cím változóban is megadható, ilyenkor a változót zárójelbe kell tenni.

CALL utasítás

Szintaxis: CALL szubrutinnév[,(paraméterlista)]

Értelmezés: Meghív egy BASIC szubrutint, melyet a SUB utasítás segítségével definiáltunk. Végrehajtás után a vezérlés visszakerül a CALL utasítást követő utasításra. A felsorolt paraméterek értéküket rendre átadják a szubrutin fejlécében szereplő változóknak.

Szubrutinnév: Egy kifejezés idézőjelek között vagy változóban, maximum 16 karakter hosszban.

Paraméterlista: Tetszés szerinti kifejezések vagy változók.

Megjegyzés: Abban az esetben, ha a paraméterek száma, ill. tipusa nem egyezik a szubrutin fejlécében szereplőkével, hibajelzést kapunk. A hibajelzésben szereplő sorszám a CALL utasítást tartalmazó sorra mutat.

Példa:

```
100 CALL "KIÍRÓ",(A,B$(0),C(0),D%)
110 ...
...
500 SUB "KIÍRÓ",(W1,W$,W2,W%)
510 PRINT W1,W$,W2,W%
520 END SUB
```

CATALOG utasítás

A CATALOG parancs azonos a DIRECTORY-paranccsal.

CBANK utasítás

Szintaxis: CBANK n[,m]

Értelmezés: Definiálja az aktuális karaktergenerátort.

n: Értéke a 0-2 intervallumban.

0: Saját karaktergenerátor ($D000-$D800 a ROM-ban).
1: Programozható karaktergenerátor ($E000-$E800 a RAM-ban).
2: Programozható karaktergenerátor ($E800-$F000 a RAM-ban).

m: Értéke a 1-2 intervallumban. Alapesetben 0.

0: Inverz megjelenítési mód bekapcsolva.

1: Inverz megjelenítési mód kikapcsolva.

Megjegyzés: Az inverz megjelenítési mód kikapcsolt állapotában 256 különböző karaktert használhatunk. Sprite-ok megjelenítése (PROJECT utasítással) ebben az üzemmódban történik, a 127-nél nagyobb kódú karaktereket felhasználva.

CENTRE utasítás

Szintaxis: CENTRE szöveg

Értelmezés: Szöveg kiírás a képernyőablak közepére.

Szöveg: maximum 40 karakter hosszú kifejezés idézőjelek között, vagy változóban(Mint PRINT-nél).

Megjegyzés: 40 karakternél hosszabb szöveg kiíratásakor STRING TOO LONG hibajelzést kapunk.

CGOTO és CGOSUB utasítások

Szintaxis: CGOTO kifejezés
CGOSUB kifejezés

Értelmezés: A CGOTO ill. CGOSUB utasítás csak annyiban tér el a GOTO ill. GOSUB utasítástól, hogy a sorszám kifejezésként is megadható.

CHANGE parancs

Szintaxis: CHANGE/szöveg1/szöveg2/

Értelmezés: Automatikus keresés és helyettesítés a programban. Csak parancsmódban használható.

: Tetszőleges elválasztó karakter (nem tartalmazza sem a keresendő, sem a helyettesítő szöveg).

Szöveg1: Keresendő szöveg.

Szöveg2: Helyettesítő szöveg.

Megjegyzés: A CHANGE parancs képes utasítások felcserélésére is, de pl. a PRINT utasítás 'N' betűjét nem lehet kicserélni.

COLLISION utasítás

Szintaxis: COLLISION n
COLLISION n[,sorszám]

Értelmezés: A legutolsó sprite kirajzolásakor (PROJECT) létrejött sprite-ütközések esetén végrahajtja a sorszámban definiált programsorban kezdődő alprogramot.

n: Értéke 1 vagy 2.

1: Sprite-sprite ütközés lekérdezése.
2: Sprite-háttér ütközés lekérdezése.

Sorszám: Egy érvényes programsorszám.

Megjegyzés: Abban az esetben, ha sorszámot nem adunk meg, törli az ütközésregiszter tartalmát.

A COLLISION utasítás csak egyszeri vezérlésátadás, nem állandó (interrupt) figyelés.

Példa 1:

```
100 COLLISION 1,200
...
200 PRINT "SPRITE-SPRITE ÜTKÖZÉS TÖRTÉNT."
210 RETURN
```

Példa 2:

```
100 COLLISION 2:REM SPRITE-HÁTTÉR ÜTKÖZÉSEK TÖRÖLVE.
```

CONCAT utasítás

Szintaxis: CONCAT [D meghajtó#1][,filenév]TO[D meghajtó#2][,filenév][ON U egységszám]

Értelmezés: Szekvenciális file tartalmának hozzáfűzése egy másik szekvenciális file-hoz.

Filenév: Egy kifejezés idézőjelek között vagy változóban, maximum 16 karakter hosszban.

Meghajtó: 0 vagy 1, attól függően melyik meghajtót kivánjuk használni. Alapesetben 0.

Egységszám: Egy egész szám a 4-15 intervallumban. Alapesetben 8.

Megjegyzés: A filenév, meghajtó és egységszám változóban is megadható, ilyenkor a változót zárójelbe kell tenni.

Példa 1: CONCAT"FILE1"TO"FILE2"

FILE1 szekvenciális file tartalmát hozzáfűzi FILE2 szekvenciális file-hoz. FILE1 tartalma változatlan marad.

Példa 2: CONCAT D(M1),(D1$) TO D(M2),(D2$)

Ebben az esetben a fileneveket és meghajtókat változókban adtuk meg.

DCLEAR utasítás

Szintaxis: DCLEAR D meghajtó[ON U egységszám]

Értelmezés: Lezárja az egységen lévő összes megnyitott csatornát (nem file-okat).

Meghajtó: 0 vagy 1, attól függően melyik meghajtót kivánjuk használni. Alapesetben 0.

Egységszám: Egy egész szám a 4-15 intervallumban. Alapesetben 8.

Megjegyzés: A meghajtó és egységszám változóban is megadható, ilyenkor a változót zárójelbe kell tenni.

DCLOSE utasítás

Szintaxis: DCLOSE [#logikai fileszám][ON U egységszám]

Értelmezés: Lezárja a megadott file-t vagy az összes megnyitott file-t.

Logikai fileszám: Egy egész szám az 1-255 intervallumban.

Egységszám: Egy egész szám a 4-15 intervallumban. Alapesetben 8.

Megjegyzés: A logikai fileszám és egységszám változóban is megadható, ilyenkor a változót zárójelbe kell tenni.

Példa 1: DCLOSE

Az összes nyitott file lezárása a 8-as lemezegység 0-ás meghajtójában lévő lemezen.

Példa 2: DCLOSE #5 ON U9

Lezárja az 5-ös logikai számú file-t a 9-es lemezegység 0-ás meghajtójában lévő lemezen.

DEFCHR utasítás

Szintaxis: DEFCHR n,v$
DEFCHR v$,n

Értelmezés: Karakter-definició elmentése változóba, vagy onnan vissza.

n: értéke 0-511 intervallumban, az aktuális karakter sorszáma.
0 -255: programozható karaktergenerátorban ($E000-$E800).
256-511: programozható karaktergenerátorban ($E800-$F000).

v$: Tetszőleges sztring típusú változó 8 karakter hosszban.

Megjegyzés: Egy karakter képe 8x8 pontból áll. Egy pont lehet ki- (0), vagy bekapcsolt (1), tehát egy pont egy biten, nyolc pont egy byte-on megadható. Egy teljes karakter definiálásához pedig nyolc byte-ra van szükség:

```
sztring 1. karakterének kódja: 76543210
sztring 2.      "         "  : 76543210
sztring 3.      "         "  : 76543210
sztring 4.      "         "  : 76543210
sztring 5.      "         "  : 76543210
sztring 6.      "         "  : 76543210
sztring 7.      "         "  : 76543210
sztring 8.      "         "  : 76543210
```

Példa 1 Egy 'A' betű definiálása:

```
---**---         16+8     = 24
--****--      32+16+8+4   = 60
-**--**- 64+32+      4+2  =102
-******- 64+32+16+8+4+2   =126
-**--**- 64+32+      4+2  =102
-**--**- 64+32+      4+2  =102
-**--**- 64+32+      4+2  =102
--------                  =  0
```

Tehát egy 'A' betű a következő képpen definiálható:

```
100 A$="":FOR I=0 TO 7:READ A:A$=A$+CHR$(A):NEXT I
110 DEFCHR A$,1
120 CBANK 1
130 PRINT "A"
140 :
150 DATA 24,60,102,126,102,102,102,0
```

Példa 2:

```
100 CBANK1
110 FOR I=0 TO 127
120 : DEFCHR I,A$
130 : FOR J=1 TO 4
140 :    A1$=MID$(A$,J,1)
150 :    MID$(A$,J,1)=MID$(A$,9-J,1)
160 :    MID$(A$,9-J,1)=A1$
170 : NEXT J
180 : DEFCHR A$,I
190 NEXT I
```

Futtassuk le a példát és egy érdekes jelenségét vehetünk észre ...

DISAPA utasítás

Szintaxis: DISAPA

Értelmezés: A program DISAPA utasítást tartalmazó sorától kezdve nem lesz listázható SECURE 10 végrehajtása után.

Példa:

```
100 REM ITT KEZDŐDIK A PROGRAM.
110 DISAPA
120 REM ITT MÉG VAN PROGRAM.
130 END

SECURE 10:LIST

100 REM ITT KEZDŐDIK A PROGRAM.
READY.
```

DOPEN utasítás

Szintaxis: DOPEN #logikai fileszám, filenév[,L hossz][,D meghajtó][,U egységszám][,W]

Értelmezés: Szekvenciális vagy relatív file megnyitása irásra vagy olvasásra.

Logikai fileszám: Egy egész szám az 1-255 intervallumban.

Filenév: Egy kifejezés idézőjelek között vagy változóban, maximum 16 karakter hosszban.

Hossz: Egy egész szám az 1-254 intervallumban, a logikai adat hossza a relatív file-ban.

Meghajtó: 0 vagy 1, attól függően melyik meghajtót kivánjuk használni. Alapesetben 0.

Egységszám: Egy egész szám a 4-15 intervallumban. Alapesetben 8.

W: Ez a paraméter megmutatja, hogy a szekvenciális file-t olvasásra vagy irásra (Write) nyitjuk meg.

Megjegyzés: Minden paraméter megadható változóban is, ilyenkor a változót zárójelbe kell tenni.

L hossz és W paraméterek egyszerre nem szerepelhetnek a DOPEN utasításban.

Példa 1: DOPEN #2,"ADATOK"

Példa 2: DOPEN #3,"CIMEK",D1,U9,W

Példa 3: DOPEN #5,"KLIENSEK",L112

DRIVE utasítás

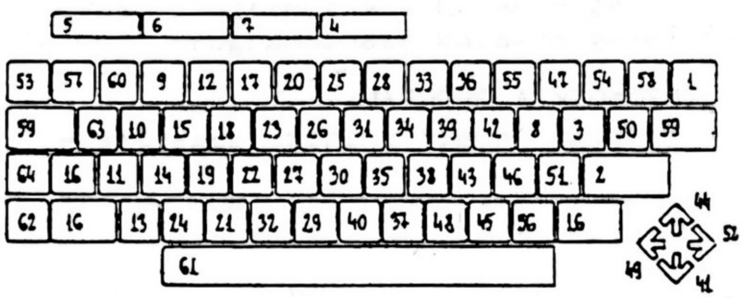
Szintaxis:
```
DRIVE ON
DRIVE OFF
DRIVE n, felfelé, lefelé, balra, jobbra
```

Értelmezés: Definiálja az n. sprite mozgatását végző billentyűket. DRIVE ON esetén be-, DRIVE OFF esetén pedig kikapcsolható a mozgatás. Csak program módban működik.

n: Az aktuális sprite száma (1-4).

felfelé, lefelé, balra, jobbra: Egy egész szám a 0-74 intervallumban, a mozgató billentyű kódja:

0: egyik billentyű sem.



Joy 1:	Joy 2:	Irány:
65	70	felfelé
66	71	lefelé
67	72	balra
68	73	jobbra
69	74	tűzgomb

Példa:
```
10 TRAP 900
100 SCNCLR:REFRESH
110 SPRITE 1,1
120 MOVSPR 1,100,100
130 SPEED 1,4,4
140 DRIVE 1,44,41,49,52
150 DO:PROJECT:LOOP
900 SPRITE CLR
910 END
```

DVERIFY utasítás

Szintaxis: DVERIFY filenév[,D meghajtó][,U egységszám]

Értelmezés: Ellenőrzi, hogy az elmentett program tartalma azonos-e a memóriában levővel.

Filenév: Egy kifejezés idezőjelek között vagy változóban maximum 16 karakter hosszban.

Meghajtó: 0 vagy 1 attól függően melyik meghajtót kivánjuk használni. Alapesetben 0.

Egységszám: Egy egész szám a 4-15 intervallumban. Alapesetben 8.

Megjegyzés: A filenév, meghajtó és egységszám változóban is megadható, ilyenkor a változót zárójelbe kell tenni.

ENVELOPE utasítás

Szintaxis: ENVELOPE n[, A][, D][, S][, R][, L][, W]

Értelmezés: Definiálja a megadható 8 burkológörbe valamelyikének paramétereit.

n: Burkológörbe száma (0-7).
A: Felfutási idő (0-15).
D: Lecsengési idő (0-15).
S: Kitartási idő (0-15).
R: Elengedési idő (0-15).
L: Kitartási szint (0-15).
W: Hullámforma (0-1).
0: négyszögjel
1: zajgenerátor

Megjegyzés: Bármelyik paraméter elhagyható, ha értékét nem kívánjuk megváltoztatni.

Hullámforma paraméter értéke is 0-15 lehet, de valójában csak a legalsó bit értéke (0 vagy 1) számít.

Bekapcsoláskor és RESET-nél a paraméterek a következő értékekre állítódnak:

n:	A	D	S	R	L	W	Hangszer
0:	0	9	0	0	8	0	Zongora
1:	12	0	12	0	8	0	Tangóharmonika
2:	0	0	15	0	8	0	Orgona 1
3:	0	5	5	0	8	0	Trombita
4:	9	4	4	0	8	0	Flöte
5:	0	9	2	1	8	0	Gitár
6:	0	9	0	0	8	0	Xylophon
7:	0	9	9	0	8	0	Orgona 2

(Lásd még a PLAY utasítás leírásánál.)

ESC utasítás

Szintaxis: ESC sztring

Értelmezés: Hatása megegyezik az ESC billentyű használatával.

szöveg: Tetszőleges kifejezés idézőjelek között vagy változóban. Az értelmezett karakterek a következők:

A: automatikus beszúrás
B: képernyő-ablak jobb alsó sarkának beállítása
C: az automatikus beszúrás törlése
D: egy sor törlése
I: egy sor utólagos beszúrása
J: ugrás a sor elejére
K: ugrás a sor végére
L: képernyőgörgetés indítása
M: képernyőgörgetés leállítása
N: az ablak törlése, visszatérés teljes képernyőre
O: beszúrás, inverz kép és villogtatás leállítása
P: törlés a sor elejétől a kurzorig
Q: törlés a kurzortól a sor végéig
R: a képernyő méretének csökkentése
T: képernyő-ablak bal felső sarkának beállítása
V: képernyőgörgetés felfelé
W: képernyőgörgetés lefelé
X: kijelölt ESC feladat leállítása

EXTEND utasítás

Szintaxis: EXTEND

Értelmezés: Bővített háttérszín üzemmód bekapcsolása.

Megjegyzés: Bővített háttérszín üzemmód esetén csak az első 64 karakter használható, de lehetőségünk van eldönteni mindegyik karakterről, hogy a beállított négy háttérszínből melyiket használja.(Ebben az üzemmódban sprite-ok, és villogó karakterek (FLASH ON) használata nem lehetséges.)

kód	b7,b6	háttérszín	cím
0-63	0 0	0. színregiszter	65301 ($FF15)
64-127	0 1	1. színregiszter	65302 ($FF16)
128-191	1 0	2. színregiszter	65303 ($FF17)
192-255	1 1	3. színregiszter	65304 ($FF18)

FAST utasítás

Szintaxis: FAST

Értelmezés: Kikapcsolja a képernyőt.

Megjegyzés: A képernyő letiltásával csökkenthető a program futásideje, ez hosszabb számítások esetén lehet hasznos. Visszakapcsolása a SLOW utasítás kiadásával lehetséges.

Példa:

```
110 TI$="000000":FOR I=1 TO 1000:NEXT:PRINT TI
RUN
 98

READY.


100 FAST:REM KÉPERNYŐ KIKAPCSOLVA
```

```
110 TI$="000000":FOR I=1 TO 1000:NEXT:PRINT TI
120 SLOW:REM KÉPERNYŐ BEKAPCSOLVA
RUN
 64

READY.
```

FETCH utasítás

Szintaxis: FETCH hossz, cím1, cím2[, érték]

Értelmezés:

Tetszőleges számú byte átmásolása a BASIC ill. KERNAL ROM-ból a RAM-ba, tetszőleges helyre.

Hossz: Egy egész szám a 0-65535 intervallumban, az átmásolandó tartomány száma.

Cím1: Egy egész szám a 0-65535 intervallumban, az első átmásolandó byte címe.

Cím2: Egy egész szám a 0-65535 intervallumban, ettől a címtől kezdődően helyezzük el az átmásolt byte-okat.

Érték: Egy egész szám a 0-255 intervallumban, az átmásolás kizáró vagy (XOR) bitmaszkja. Alapesetben 0.

Példa: FETCH DEC("0800"),DEC("D000"),DEC("E000")

A PLUS/4 saját karaktergenerátorát az első programozható karakterkészlet helyére másolja.

FILL utasítás

Szintaxis: FILL hossz, cím, érték

Értelmezés: A memória feltöltése adott hosszban, adott címtől kezdve, adott értékkel.

Hossz: Egy egész szám a 0-65535 intervallumban.

Cím: Egy egész szám a 0-65535 intervallumban.

Érték: Egy egész szám a 0-255 intervallumban, a feltöltő byte értéke.

Példa: FILL 1000,3072,1

Feltölt 3072 címtől kezdve 1000 byte-ot 1-gyel.

FIND parancs

Szintaxis: FIND /szöveg/

Értelmezés: Kilistázza a program azon sorait, amiben a keresendő szöveg, szerepel. Csak parancsmódban használható.

/: Tetszőleges elválasztó karakter (nem tartalmazza a keresendő szöveg).

Szöveg: A keresendő szöveg.

Megjegyzés: Csak parancs üzemmódban használható.

FRAME utasítás

Szintaxis: FRAME szöveg

Értelmezés: A WOPEN parancs ablakkeret-karaktereit, és feltöltő karakterét definiálja.

Szöveg: Egy 9 karakter hosszú kifejezés idézőjelek között vagy változóban.

Példa:

```
100 FRAME "123456789"
110 WOPEN 1,1,6,6,3
120 :
130 FRAME "ABCD FGHI"
140 WOPEN 9,9,14,14,3
```

A képernyőn megjelenő ablakok a következők:

```
122223  ABBBBC
455556  D    F
455556  D    F
455556  D    F
455556  D    F
788889  GHHHHI
```

GETIN utasítás

Szintaxis: GETIN xhely, yhely, hossz, szöveg, változó

Értelmezés: Ellenőrzött input a képernyő tetszőleges helyéről, korlátozott hosszban, adott karakterekkel. Csak programmódban használható.

Xhely: Kurzor x kezdőpozició.

Yhely: Kurzor y kezdőpozició.

Hossz: Inputsor maximális hossza.

Szöveg: Beírható karakterek (üres sztring megadása esetén a GETIN utasítás minden karaktert elfogad).

Megjegyzés: A RETURN és DEL billentyűket minden esetben elfogadja, a RETURN az inputsor végét jelenti, a DEL billentyűvel az inputsor legutolsó karakterét törölhetjük.

HARDCOPY utasítás

Szintaxis: HARDCOPY [U egységszám]

Értelmezés: A karakteres vagy grafikus képernyő tartalmát kiírja az adott egységre.

Egységszám: Egy egész szám a 4-15 intervallumban. Alapesetben 4.

Megjegyzés: Az egységszám változóban is megadható, ilyenkor a változót zárójelbe kell tenni.

A HARDCOPY utasítás mindig az éppen látható képernyő tartalmát nyomtatja ki. Karakteres képernyő kinyomtatásánál a

programozható karaktereket, sprite-okat is kinyomtatja, de grafikusan osztott képernyő is kinyomtatható vele.

Példa 1:
```
100 GRAPHIC 0
110 CBANK 1
120 HARDCOPY
```

Kinyomtatja a karakteres képernyő tartalmát az első programozható karaktergenerátor karaktereivel

Példa 2:
```
100 GRAPHIC 1
110 HARDCOPY U5
120 GRAPHIC 0
```

Kinyomtatja a grafikus képernyő tartalmát az 5-ös egységszámú nyomtatóra.

INBOX utasítás

Szintaxis: INBOX n, x1, y1, x2, y2, mód THEN ...

Értelmezés: Abban az esetben, ha az n. sprite az x1,y1 és x2,y2 által meghatározott téglalapon belül található, végrehajtódnak a THEN után álló további utasítások.

n: Az aktuális sprite száma (1-8).

x1, y1: Téglalap bal felső sarka karakteresen.

x2, y2: Téglalap jobb alsó sarka karakteresen.

Mód: Téglalap által bezárt karakterek invertálása:
0: a téglalapot nem forgatja inverzbe.
1: a téglalapot inverzbe forgatja.

Példa:
```
100 SPRITE 1,1
110 MOVSPR 1,100,100
120 DO
130 : MOVSPR 1,+0,-1
140 : PROJECT
150 : INBOX 1,0,0,39,0,0 THEN EXIT
160 LOOP
170 PRINT "A SPRITE A LEGFELSÖ SORBA ÉRT"
180 END
```

INKEY függvény

Szintaxis: f=INKEY (n)

Értelmezés: Az INKEY függvény argumentuma az éppen lenyomott funkcióbillentyű sorszáma.

n: Tetszés szerinti egész szám a 0-255 intervallumban, érték nem befolyásolja az INKEY függvény müködését.

Megjegyzés: Az INKEY függvény értéke 0-16 intervallumba esik:

0: nincs lenyomott funkcióbillentyű.
1: F1 <F1>
2: F2 <F2>
3: F3 <F3>
4: F4 <SHIFT+F1>
5: F5 <SHIFT+F2>

```
 6:   F6  <SHIFT+F3>
 7:   F7  <SHIFT+HELP>
 8:   F8  <HELP>
 9:   F9  <CTRL+F1>
10:  F10  <CTRL+F2>
11:  F11  <CTRL+F3>
12:  F12  <C=+F1>
13:  F13  <C=+F2>
14:  F14  <C=+F3>
15:  F15  <C=+HELP>
16:  F16  <CTRL+HELP>
```

JUMP utasítás

Szintaxis: JUMP szubrutinnév[, (paraméterlista)]

Értelmezés: Feltétel nélküli vezérlésátadás a szubrutinnév címkével megjelölt (SUB utasítás) programsorra.

Szubrutinnév: Egy kifejezés idezőjelek között vagy változóban maximum 16 karakter hosszban.

Paraméterlista: Tetszés szerinti állandók és változók.

Megjegyzés: Abban az esetben, ha a felsorolt változók száma, ill. tipusa nem egyezik a szubrutin fejlécében szereplőkével, hibajelzést kapunk. A hibajelzésben szereplő sorszám a JUMP utasítást tartalmazó sorra mutat.

NORMAL utasítás

Szintaxis: NORMAL

Értelmezés: Karakteres alapüzemmód bekapcsolása kiadott EXTEND- vagy MULTI utasítás után.

MERGE utasítás

Szintaxis: MERGE filenév[,D meghajtó][,U egységszám]

Értelmezés: Hozzáfűz egy program file-t a memóriában levő BASIC programhoz.

Filenév: Egy kifejezés idezőjelek között vagy változóban maximum 16 karakter hosszban.

Meghajtó: 0 vagy 1 attól függően melyik meghajtót kivánjuk használni. Alapesetben 0.

Egységszám: Egy egész szám a 4-15 intervallumban. Alapesetben 8.

Megjegyzés: A filenév, meghajtó és egységszám változóban is megadható, ilyenkor a változót zárójelbe kell tenni.

FIGYELEM! Ha a tárban levő programban van a ráfűzendő program legkisebb sorszámánál nagyobb sorszám, akkor hibás lesz az összefűzés.

MOVE utasítás

Szintaxis: MOVE x1, y2, x2, y2, x3, y3[, mód]

Értelmezés: Az x1, y1 és x2, y2 által meghatározott téglalapot átmásolja x3,y3 karakterhelytől kezdődően.

x1, y1: A téglalap bal felső csúcsának koordinátái karakteresen.

x2, y2: A téglalap jobb alsó csúcsának koordinátái karakteresen.

x3, y3: Az átmásolt téglalap bal felső csúcsának koordinátái karakteresen.

mód: Az átmásolás módja. (Alapesetben 3.)
- 0: nem történik másolás.
- 1: csak a színmemóriát másolja át.
- 2: csak a képmemóriát másolja át.
- 3: szín- és képmemóriát is átmásolja.

MOVSPR utasítás

Szintaxis: MOVSPR n, x, y
MOVSPR n, +/- dx, +/- dy
MOVSPR n, l; szög
MOVSPR n, sebesség # szög
A

Értelmezés: Sprite mozgatása a megadott helyre, elmozdítása a megadott értékekkel, vagy mozgási sebességének és irányának beállítása. A folyamatos mozgatásról megszakításból hívott rutin gondoskodik.

n: Az aktuális sprite száma (1-8).

x, y: Abszolút koordináták, a sprite koordinátái.

+/- dx, +/- dy: Relatív koordináták, az eltolás nagysága x és y irányba.

l: Relatív távolság, az eltolás nagysága.

sebesség: A sprite sebessége (0-15).

szög: A sprite elmozdulásának iránya fokokban megadva (0-360).

Megjegyzés: A sprite-ok láthatósági területe az aktuális képrenyőablak területére megegyezik.

Teljes képernyő esetén:

```
 (0,0)
*-------------------------->
!
!  (24,24)          (344,24)
!  *-------------------*
!  !                   !
!  !                   !
!  !                   !
!  !                   !
!  !                   !
!  !       KÉPERNYŐ    !
!  !                   !
!  !                   !
```

```
!  !                 !
!  !                 !
!  !(24,224)         !(344,224)
!  *-----------------*
!
!
!
```

Sprite koordinátáin a sprite bal felső sarkának koordinátái értendők. A sprite méreti vízszintesen 24, függőlegesen 21 képpont.

A relatív koordináták egyszeri elmozdulást jelentenek, minden kirajzoláskor ´lép´ egyet a sprite.

A bekapcsolt sprite-okat PROJECT utasítással írhatjuk ki a képernyőre.

A koordináták változtatását DRIVE ON parancs engedélyezi és DRIVE OFF parancs tiltja le.

MULTI utasítás

Szintaxis: MULTI

Értelmezés: Többszínű üzemmód bekapcsolása.

Megjegyzés: Bekapcsolt többszínű üzemmód esetén a szinmemóriában felemelt 7. bit jelöli a többszínű karaktert. Az így definiált karakterben a vízszintes felbontás a felére csökken, és az összetartozó bitpárok a következő színekben látszanak:

00 bitpár: háttérszín (COLOR 0)
01 bitpár: többszín #1 (COLOR 2)
10 bitpár: többszín #2 (COLOR 3)
11 bitpár: karakterszín (COLOR 1)

OLD utasítás

Szintaxis: OLD

Értelmezés: Hatástalanítja a NEW parancsot.

Megjegyzés: OLD parancs RESET után is használható a program visszanyerésére

ONKEY utasítás

Szintaxis: ONKEY szöveg THEN ... ELSE ...

Értelmezés: Ellenőrzi, hogy valamelyik a sztringben szereplő billentyűt lenyomtunk-e. Ha igen, a vezérlésátadás végrehajtódik.

szöveg: kifejezés idézőjek között vagy változóban megadva.

OPTION utasítás

Szintaxis: OPTION n

Értelmezés: Ha n értéke 10, az ezt követően kiadott LIST utasítások a PLUS/128 BASIC bővítés alapszavait inverz formában írják ki a

képernyőre, vagy nyomtatóra.

Ha a paraméter értéke ettől eltérő (0-255), akkor a listázás normális.

PAGE utasítás

Szintaxis: PAGE n

Értelmezés: Az utasítás megadja, hogy a LIST hányasával listázza ki a programsorokat. n értékét 0-nál nagyobbra vállasztva a LIST hatására csak az első n programsor jelenik meg a képernyőn. A SZóKöZ lenyomásával a listázás tovább folytatható n soronként, a STOP lenyomásával pedig bárhol megallítható. PAGE 0 a normális listázást allítja vissza.

PATTERN utasítás

Szintaxis: PATTERN n
PATTERN n, vonalminta
PATTERN n, területminta, területminta,..., területminta
PATTERN , területminta, területminta,..., területminta

Értelmezés: Definiál egy vonal- vagy kitöltési mintát a grafikus utasítások számára.

n: Egy egész szám a 0-15 intervallumban a definiálandó minta sorszáma.

Vonalminta: Egy egész szám a 0-65535 intervallumban, a vonalrajzolás 16 bites maszkja.

Területminta: m darab egész szám a 0-65535 intervallumban a területfeltöltés 16 bit széles és m bit magas maszkja. A területminta több egymásutáni PATTERN utasítással is megadható (pl. nem férne el egy sorban), ilyenkor az elsőt követő PATTERN utasításokból kihagyandó a vonalminta sorszáma.

Példa 1:

```
100 GRAPHIC 1,1
110 PATTERN 1
120 DRAW 1,10,10 TO 310,10
```

Példa 2:

```
100 GRAPHIC 1,1
110 PATTERN 1,DEC("0FFF")
120 DRAW 1,10,10 TO 310,10
```

Példa 3:

```
100 GRAPHIC 1,1
110 V$=DEC("0FFF")
120 PATTERN 1,V$,V$
130 DRAW 1,10,10 TO 310,10
```

Példa 4:

```
100 GRAPHIC 1,1
110 PATTERN 8,DEC("F000"),DEC("0F00"),DEC("00F0"),DEC("000F")
120 BOX 8,10,10,100,100,,1
```

PEEK$ függvény

Szintaxis: v$=PEEK$ (cím, hossz)

Értelmezés: Elhelyezi sztringváltozóban a memóriában byte-onként tárolt adatot.

Cím: a kiolvasás kezdőcíme (0-65535).

Hossz: a kiolvasás hossza (0-255).

Példa 1:
```
100 POKE 3072,"EZ"," EGY"," ÚJDONSÁG!!!"110 A$=PEEK$(3072,15)
120 PRINT A$
```

PLAY utasítás

Szintaxis: PLAY szöveg

Értelmezés: Lejátsza a szövegben definiált dallamot.

szöveg Tetszőleges sztring típusú változó. Az értelmezett karakterek az alábbiak:

On: Oktáv beállítása (n=0 - n=4).
Tn Beállított burkológörbe kiválasztása (n=0 - n=7).
Un: Hangerő beállítása (n=0 - n=8).
N: Egy hang (A,B,C,D,E,F,G).
#N: Félhang állítás felfelé.
$N: Félhang állítás lefelé.
W: A következő hangok egész hangok.
H: A következő hangok félhangok.
Q: A következő hangok negyedek.
I: A következő hangok nyolcadok.
S: A következő hangok tizenhatodok.
.N: A következő hang pontozott (A felével hosszabb).
R: Egy szünetet tart.
M: Várakozás az utolsó hang teljes lefutására.

Példa:
```
100 ENVELOPE 1,0,9,2,1,8,0:PLAY"T1"
110 VOL 8
120 TEMPO 100
130 PLAY "CDEFGAB"
140 PLAY "U402CDEFGAB"
150 PLAY "ICCCSCCC"
160 PLAY "U803.HCDQ#EFI$GA.S#B"
```

A 100-as sorban beállítjuk az első burkológörbét, majd kiválasztjuk.
A 110-es sorban beállítjuk a hangerőt.
A 120-as sorban beállítjuk a tempót.
A 130-as sorban lejátszunk C D E F G A B hangokat (hangskála).
A 140-es sorban is lejátszuk az előbbi hangokat, de előbb 4-esre állítjuk a hangerőt és 2-esre az oktávot.
A 150-es sorban nyolcadokat, majd tizenhatodokat.
A 160-es sort lejátszva pontozott ritmust, félhang le- és felváltást is hallhatunk.

POINTER függvény

Szintaxis: v=POINTER (változónév)

Értelmezés: Változó címe a memóriaban. Értéke a 0-65535 intervallumban lehet.

Megjegyzés: A POINTER utasítás a nem definiált változókat definiálja.

Példa 1:
```
100 BX%=1024
110 VP=POINTER(BX%)
120 AX=256*PEEK(VP)+PEEK(VP+1)
130 PRINT AX
```

Példa 2:

```
100 WORD 55,DEC("8000"):CLR
110 A$="":FOR I=0 TO 5:READ B$:A$=A$+CHR$(DEC(B$)):NEXT I
120 A=POINTER(A$)
130 SYS(PEEK(A+1)+256*PEEK(A+2))
140 DATA A0,00,98,99,00,0C,C8,D0,F9,60
```

PROFF és PRON utasítások

Szintaxis: PROFF
PRON

Értelmezés: Ki- vagy bekapcsolja a STOP billentyű figyelését.

Megjegyzés: PRON utasítást követően futó programot csak a RESET-gomb megnyomásával lehet megszakítani.

PROJECT utasítás

Szintaxis: PROJECT

Értelmezés: Kivetíti az előzőleg definiált, bekapcsolt sprite-okat.

Példa:

```
100 SPRITE 1,1,,,100
110 MOVSPR 1,100,100
120 PROJECT
```

PUFF utasítás

Szintaxis: PUFF n

Értelmezés: A PUFF utasítás n értékenek megfelelő hosszúra állítja a billentyűzet-puffer hosszát

n: A puffer hossza (1-10). Alapesetben 10.

PUFF$ utasítás

Szintaxis: PUFF$ szöveg

Értelmezés: A szöveget beírja a billentyűzet-pufferbe.

szöveg: Minimum egy, maximum billentyűzetpuffer-hossz hosszúságú kifejezés idézőjelek között vagy változóban.

RCBANK függvény

Szintaxis: v=RCBANK (n)

Értelmezés: Az aktuális karaktergenerátor lekérdezése.

Értéke n=0 esetén:

0: A karaktergenerátor a ROM-ban van ($D000-$D7FF)
1: A karaktergenerátor a RAM-ban $E000-$E7FF címen
2: A karaktergenerátor a RAM-ban $E800-$EFFF címen
3: A karaktergenerátor a RAM-ban van, de nem $E000-$EFFF címen

Értéke n=1 esetén:

0: Inverz megjelenítés bekapcsolva.
1: Inverz megjelenítés kikapcsolva.

Példa 1:

```
100 WORD 55,DEC("D800"):CLR
110 FETCH DEC("0800"),DEC("D000"),DEC("D800")
120 CBANK 1
130 POKE 2043,DEC("D8")
```

Példa 2:

```
100 WORD 55,DEC("D800"):CLR
110 BLOAD "KÉSZLET",P(DEC("D800"))
120 CBANK 1
130 POKE 2043,DEC("D8")
```

A példák bemutatják, hogyan lehet saját programozható karaktergenerátort használni akár a $E000-$F000 memóriarészen kívül eső területeken.
Mindkét példa magyarázata:
A 100-as sorban letiltjuk a BASIC változók elől a $D800-$E000 területet (Alapesetben az 55,56 címen $E000 mutató található).
A 110-es sorban előállítjuk a készletet, akár másolással a ROM-ból, akár betöltéssel, lemezről.
A 120-as sorban átkapcsolunk a programozható karaktergenerátorok valamelyikére.
A 130-as sorban átírjuk a programozható karaktergenerátorok mutatóját úgy, hogy az a saját készletünkre mutasson.

RCOMP utasítás

Szintaxis: RCOMP THEN ... ELSE ...

Értelmezés: A legutóbbi IF...THEN...ELSE feltétele szerint hajt végre elágazást, vagy sem.

Példa:

```
100 INPUT A
110 IF A>0 THEN PRINT "POZITíV
120 A=-1
130 RCOMP THEN PRINT "POZITíV
```

RECORD utasítás

Szintaxis: RECORD #logikai fileszám, rekord szám[, pozició]

Értelmezés: Az RECORD utasítás a megadott számu relatív file iró/olvasó fejét az adott rekord megfelelő pozíciójú helyére állítja.

logikai fileszám: Egy egész szám az 1-255 intervallumban

rekordszám: Egy egész szám az 1-65535 intervallumban.

pozició: Egy egész szám az 1-254 intervallumban. Alapesetben 1.

Megjegyzés: A logikai fileszám, rekordszám és pozíció változóban is megadható, ilyenkor a változót zárójelbe kell tenni.

Nemlétező helyre pozícionálásnál RECORD NOT PRESENT hibaüzenetet kapunk.

Példa: RECORD #2,452,23
Az írófejet a már megnyitott kettes logikai számú relatív file 452. rekordjának 23. byte-jára állítja.

REFRESH utasítás

Szintaxis: REFRESH

Értelmezés: A sprite-rajzolo rutin tudtára adjuk, hogy időközben új képernyőt rajzoltunk, és nem akarjuk, hogy a következő sprite-kivetítéskor (PROJECT) visszaírja a sprite-ok alatti, általunk már letörölt karaktereket.

Példa:

```
100 SPRITECLR:SCNCLR
110 GOSUB 5000:REM ÚJ KEPERNYŐ KIRAJZOLÁSA
120 REFRESH
130 ...
```

RESET utasítás

Szintaxis: RESET

Értelmezés: Az OCTASOFT BASIC újraindítása. Hatására a PLUS/4-es végrehajtja a teljes reset ciklust.

RKEY függvény

Szintaxis: v=RKEY (n)

Értelmezés: Megadja az éppen lenyomott billentyű kódját, vagy megmondja, hogy a kérdéses billentyű éppen le van-e nyomva. Az RKEY függvény segítségével egyszerre több billentyű lenyomását is figyelemmel kisérhetjük a programból.

A billentyűk kódkiosztása a DRIVE utasítás leírásában található.

Értéke n=0 esetén:
Ha nincs lenyomva egy billentyű sem 0, ellenkező esetben az éppen lenyomott billentyű kódja.
0 < n < 75 esetén pedig:
Ha a kérdéses kódú billentyű nincs lenyomva 0, ellenkező esetben pedig 1.

Példa:

```
100 SCNCLR
110 X=10:Y=10
120 CHAR,X,Y,"*"
130 X=X-(RKEY(49)ANDX>1)+(RKEY(52)ANDX<38)
140 Y=Y-(RKEY(44)ANDY>1)+(RKEY(41)ANDY<23)
150 CHAR,X,Y,"<CTRL+9>*<CTRL+0>"
160 GOTO 120
```

Ez egy egyszerű rajzolóprogram. Rajzolás a kurzormozgató billentyükkel történik. Ha egyszerre nyomjuk le pl. a jobb és felfele gombot akkor mindkét irányba lép átlósan. Ez RKEY nélkül nem oldható meg ilyen egyszerűen.

RREG utasítás

Szintaxis: RREG [A változó][, X változó][, Y változó][, S változó]

Értelmezés: Egy SYS utasításban meghívott gépikódú-alprogram visszatérése után az A,X,Y processzorregiszterek es SR státuszregiszter

beolvasása BASIC változókba.

A változó, X változó, Y változó, S változó: tetszésszerinti változók.

RSBANK függvény

Szintaxis: v=RSBANK (n)

Értelmezés: Az aktuális sprite-bank lekérdezése. Értéke 0-3 intervallumba esik.

0: Sprite-definíciók a $0000-$3FFF területen találhatók.
1: Sprite-definíciók a $4000-$7FFF területen találhatók.
2: Sprite-definíciók a $8000-$BFFF területen találhatók.
3: Sprite-definíciók a $C000-$FFFF területen találhatók.

n: Tetszőleges egész szám a 0-255 intervallumban, értéke nem befolyásolja RKEY függvény értékét.

RSPCOLOR függvény

Szintaxis: v=RSPCOLOR (n)

Értelmezés: Megadja az aktuális sprite színkódját (Alapállapotban minden sprite-é 1).

n: A sprite száma (1-8).

Megjegyzés: A 16-os színt az RCLR függvényhez hasonlóan 0-ásnak írja ki.

RSPLUM függvény

Szintaxis: v=RSPLUM (n)

Értelmezés: Megadja az aktuális sprite fényességkódját. (Alapállapotban minden sprite-é 0).

n: A sprite száma (1-8).

RSPPOS függvény

Szintaxis: v=RSPPOS (n, m)

Értelmezés: Megadja az aktuális sprite helyzetét, sebességét.

n: A sprite száma (1-8).

m: Egy egész szám a 0-5 intervallumban.

0: Sprite x pozíció.
1: Sprite y pozíció.
2: Sprite x pozíció karakteresen.
3: Sprite y pozíció karakteresen.
4: Sprite x irányú sebessége.
5: Sprite y irányú sebessége.

RSPRITE függvény

Szintaxis: v=RSPRITE (n, m)

Értelmezés: Megadja az aktuális sprite paramétereit.

n: A sprite száma (1-8).

m: Egy egész szám a 0-11 intervallumban.

0: Értéke 1, amikor a sprite be- és 0, amikor kikapcsolt.
1: Értéke 0 amikor a sprite prioritása nagyobb, mint a háttéré, különben 1.
2: Értéke 1 amikor a sprite többszínű, különben 0.
3: Az aktuális sprite-lap száma (0-255)
4: A felfelé mozgató billentyű kódja (0-74).
5: A lefele mozgató billentyű kódja (0-74).
6: A jobbra mozgató billentyű kódja (0-74).
7: A balra mozgató billentyű kódja (0-74).
8: A megengedett maximális x irányú sebesség (0-31).
9: A megengedett maximális y irányú sebesség (0-31).
10: A beállított gyorsulási idő (0-31).
11: A beállított lassulási idő (0-31).

RWINDOW függvény

Szintaxis: v=RWINDOW (n)

Értelmezés: Megadja az aktuális képernyőablak paramétereit.

0: Értéke az ablak bal felső sarkának x koordinátája.
1: Értéke az ablak bal felső sarkának y koordinátája.
2: Értéke az ablak jobb alsó sarkának x koordinátája.
3: Értéke az ablak jobb alsó sarkának y koordinátája.

SBANK utasítás

Szintaxis: SBANK n

Értelmezés: Definiál 16-kbyte memóriaszeletet a sprite-definiciók számára.

n: Értéke 0-3 intervallumban.

0: Sprite-definíciók a $0000-$3FFF területen.
1: Sprite-definíciók a $4000-$7FFF területen.
2: Sprite-definíciók a $8000-$BFFF területen.
3: Sprite-definíciók a $C000-$FFFF területen.

SCROLL utasítás

Szintaxis: SCROLL x1, y1, x2, y2, irány[, típus][, mód]

Értelmezés: A képernyő megadott részét az adott irányban görgeti.

x1, y1: A görgetendő téglalap bal felső sarka.

x2, y2: A görgetendő téglalap jobb alsó sarka.

Irány: A görgetés iránya (mint a JOY függvény értékei):
0: Nem történik görgetés
1: Görgetés felfelé
2: Görgetés felfelé és jobbra
3: Görgetés jobbra
4: Görgetés lefelé és jobbra
5: Görgetés lefelé

6: Görgetés lefelé és balra
7: Görgetés balra
8: Görgetés felfelé és balra

Típus: 1 megadása esetén a kilépő karakterek az ellenkező oldalon visszatérnek, 0 esetén nem. Alapesetben 0.

Mód: A görgetés módja. Alapesetben 3.
0: Nincs változás
1: Csak a színmemória
2: Csak a képmemória
3: Szín- es képmemória együttesen

Megjegyzés: Ha bekapcsolt sprite-oknál használjuk a SCROLL utasítást, akkor a görgetés elött a sprite-okat letörli, majd görgetés után újra kirajzolja azokat.

Példa:

```
100 DO:SCROLL 1,1,10,10,JOY(1),1:LOOP
```

Az 1. joystick elmozdításának irányába görgeti a képernyőt. Nem kapunk ILLEGAL QUANTITY hibajezést a tűzgomb lenyomásakor, mert az irány paraméter beolvasásakor a 7. bitet nem veszi figyelembe!

SECURE utasítás

Szintaxis: SECURE n

Értelmezés: SECURE 10 utasítás végrehajtása után nem listázható a program attól a sortól kezdődően, melyben DISAPA utasítás található.

n: Tetszőleges egész szám a 0-255 intervallumban. A SECURE utasítás n=10 esetén hatásos, különben nem.

SIZE utasítás

Szintaxis: SIZE filenév[,D meghajtó][,U egységszám][,P cím]

Értelmezés: Adatfile töltési végcímének meghatározása tetszőleges címtől kezdődően. A kezdő- és végcím hexadecimális formában kerül kiírásra.

Filenév: Egy kifejezés idézőjelek között vagy változóban, maximum 16 karakter hosszban.

Meghajtó: 0 vagy 1, attól függően melyik meghajtót kivánjuk használni. Alapesetben 0.

Egységszám: Egy egész szám a 4-15 intervallumban. Alapesetben 8.

Cím: Egy egész szám a 0-65535 intervallumban, a betöltés kezdőcíme. Amennyiben nincs megadva, úgy a kimentési címtől kezdődően történik a számítás.

Megjegyzés: A filenév, meghajtó, egységszám és cím változóban is megadható, ilyenkor a változót zárójelbe kell tenni.

Példa:

```
SIZE "KÉP"
SEARCHING FOR 0:KÉP
$0800-$0FFF
READY.
```

Az adatfile töltési végcíme $0FFF (4095).

SLEEP utasítás

Szintaxis: SLEEP n, szín, x, y, szöveg, mód

Értelmezés: Kiírja a szöveget a képernyő megfelelő helyére (mint a CHAR utasítás) és a program futását felfüggeszti n értékének megfelelő ideig, illetve a SZÓKÖZ-billentyű lenyomására vár.

n: Egy egész szám a 0-65535 intervallumban, a várakozási idő másodpercben. 0 megadásánál SLEEP utasítás a SZÓKÖZ-billentyű lenyomására vár.

szín: Egy egész szám a 0-3 intervallumban, alapesetben 1.

x: Kezdő oszlopszám (0-39).

y: Kezdő sorszám (0-24).

Szöveg: A kiírandó szöveg idézőjelek között vagy változóban.

Mód: 0 vagy 1 aszerint, hogy normál vagy inverz a kiírás.

SLOW utasítás

Szintaxis: SLOW

Értelmezés: Bekapcsolja a képernyőt.

SPEED utasítás

Szintaxis: SPEED n, x[, y][, a][, b]

Értelmezés: Definiálja az n.-ik sprite mozgatási paramétereit.

n: A sprite száma (1-8).

x: A sprite megengedett legnagyobb x irányú sebessége (0-31).

y: A sprite megengedett legnagyobb y irányú sebessége (0-31).

a: A sprite gyorsulási időállandója (0-31).

b: A sprite lassulási időállandója (0-31).

SPRCOLOR utasítás

Szintaxis: SPRCOLOR n, szín[, fényesség]

Értelmezés: Definiálja az aktuális sprite színét.

n: A sprite száma (1-8).

Szín: Egy egész szám az 1-16 intervallumban.

Fényesség: Egy egész szám a 0-7 intervallumban. Alapesetben 7.

SPRDEF utasítás

Szintaxis: SPRDEF

Értelmezés: Bekapcsolja a sprite-editort

Sprite-editor funkciók:

A: Automatikus továbbléptetés funkció ki- és bekapcsolása.

Kurzorvezérlő gombok: Kurzor mozgatása a szerkesztő mezőben.

RETURN: Kurzor állítása a következő sor elejére a szerkesztő mezőben.

ESC: Kilépés az editor funkcióból.

HOME: Kurzor állítása a szerkesztő mező bal felső sarkába.

CLR: Szerkesztő mező törlése.

1, 2, 3, 4: Képpont írása a kurzor helyén. 1 esetén háttérszínű, alap üzemmódban 2,3,4 esetén sprite-színű, többszínű üzemmódban 2-nél többszín#1, 3-nál többszín#2 4-nél sprite-színű pont írása.

<CTRL + 1> - <CTRL + 8>: Sprite-szín választás 1-8-ig.

<C= + 1> - <C= + 8>: Sprite-szín választás 9-16-ig.

M: Sprite többszín üzemmód ki- és bekapcsolása.

+: Sprite-lap váltása felfelé (127-ig)

-: Sprite-lap váltása lefelé (0-ig)

Megjegyzés: Sprite szerkesztéskor a $C000-$DFFF memóriaterületen levő (SBANK3-ban 0-127-ig) sprite-lapok szerkeszthetők meg. Bekapcsoláskor ez a terület BASIC-program és változók számára szabadon használható terület egészen $E000-ig. A BASIC előli letiltása a következőképpen lehetséges:

```
10 WORD 55,DEC("C000"):CLR
```

SPRITE utasítás

Szintaxis: SPRITE CLR
SPRITE OFF
SPRITE n,[akt],[pri],[mód],[sprite-lap]

Értelmezés: Sprite-ok letörlése a képernyőről, sprite-ok kikapcsolása, vagy a sprite paraméterek megadása.

n: A sprite száma (1-8).

Akt: 0 vagy 1, aszerint, hogy a sprite-ot ki- (0) vagy bekapcsoljuk (1).

Pri: 0 vagy 1, aszerint, hogy a sprite prioritása nagyobb (0) vagy kisebb (1),mint a háttéré.

Mód: 0 vagy 1, aszerint, hogy a sprite tobbszínu (1) vagy nem (0).

Sprite-lap: Egy egész szám 0-255 intervallumban, aszerint,

hogy a sprite definíciója a lehetséges 256 sprite-lap közül melyiken található.

Megjegyzés: A nem változtatandó paraméterek elhagyhatók.

Többszínűnek definiált sprite-ok a MULTI parancs kiadása esetén válnak többszínűvé.

Példa:

```
100 REFRESH
110 SPRITE 4,1
120 MOVSPR 4,100,100
130 PROJECT
```

SPRSAV utasítás

Szintaxis: SPRSAV n, v$
SPRSAV v$, n

Értelmezés: Sprite-definíció elmentése sztring változóba vagy onnan vissza.

n: A sprite száma (1-8).

v$: Egy 63 karakter hosszú kifejezés idézőjelek között vagy változóban.

Példa:

```
100 A$="":FOR I=0 TO 62:READ A:A$=A$+CHR$(A:NEXT I
110 SPRSAV A$,1
...
1000 DATA ... :REM SPRITE ADATOK, 63 BYTE
```

Sprite definíciója data sorokban van letárolva. Az egyes adatok a következő módon alkotják a sprite-ot:

```
 0. byte    1. byte    2. byte
 3. byte    4. byte    5. byte
 6. byte    7. byte    8. byte
   ...        ...        ...
   ...        ...        ...
   ...        ...        ...
60. byte   61. byte   62. byte.
```

STASH utasítás

Szintaxis: STASH hossz, cím1, cím2[, érték]

Értelmezés: Tetszőleges számú byte átmásolása a RAM memóriában tetszőleges helyre.

Hossz: Egy egész szám a 0-65535 intervallumban, az átmásolandó tartomány hossza.
Cím1: Egy egész szám a 0-65535 intervallumban, az első átmásolandó byte címe.

Cím2: Egy egész szám a 0-65535 intervallumban, ettől a címtől kezdődően helyezzük el az átmásolt byte-okat.

Érték: Egy egész szám a 0-255 intervallumban, az átmásolás kizáró vagy (XOR) bitmaszkja. Alapesetben 0.

Példa:

```
100 STASH 480,3072,3552
```

SUB utasítás
END SUB utasítás
EXIT SUB utasítás

Szintaxis:
```
SUB szubrutinnév[,(változólista)]
END SUB [szubrutinnév]
EXIT SUB [szubrutinnév]
```

Értelmezés: A SUB utasítás a programsort a szubrutinnév címkével látja el. Az END SUB utasítás jelenti a szubrutin végét. Végrehajtása után a vezérlés visszakerül CALL utasítás utáni első utasításra.
Az EXIT SUB utasítás félbeszakítja a szubrutin végrehajtását és visszaadja a vezérlést a CALL utasítás utáni első utasításra.

Szubrutinnév: Egy kifejezés idézőjelek között maximum 16 karakter hosszban.

Változólista: Tetszés szerinti változók.

Megjegyzés: A SUB utasításnak a programsor első utasításának kell lennie.

Példa:
```
100 SUB "OLVASÓ"
110 GETKEY E$
120 IF E$="<CTRL+Q>" THEN EXIT SUB
130 GETKEY F$
140 END SUB
```

SWAP utasítás

Szintaxis:
```
SWAP hossz, cím1, cím2[, érték]
```

Értelmezés: Memóriarészek felcserélése a RAM memóriában.

Hossz: Egy egész szám a 0-65535 intervallumban, a felcserélendő hosszúság.

Cím1: Egy egész szám a 0-65535 intervallumban, az első felcserélendő byte címe.

Cím2: Egy egész szám a 0-65535 intervallumban, az első felcserélendő byte címe.

Érték: Egy egész szám a 0-255 intervallumban, a felcserélés kizáró vagy (XOR) bitmaszkja.

Példa:
```
100 SWAP 1000,52000,3072
110 GETKEY E$
120 SWAP 1000,52000,3072
```

Felcserél 1000 byte-ot 52000 és 3072 címektől, vár egy billentyű lenyomására, majd visszacseréli az 1000 byte-ot.

TEMPO utasítás

Szintaxis:
```
TEMPO n
```

Értelmezés: Definiálja a PLAY utasítás zene lejátszási sebességét.

n: Egy egész szám a 0-255 intervallumban.

Példa:
```
100 TEMPO 100
110 PLAY "CDEFGAB"
```

```
120 TEMPO 150
130 PLAY "CDEFGAB"
```

WCLOSE utasítás

Szintaxis: WCLOSE

Értelmezés: A WOPEN utasítással ellmentett képernyő-téglalap visszaírása a képernyőre.

Példa:

```
100 WOPEN 0,0,39,24,1
110 GETKEY E$
120 WCLOSE
```

Elmenti a teljes képernyő tartalmát, letörli a képernyőt, vár egy billentyű lenyomására, majd visszaírja a kiindulási képernyő tartalmát.

WFILL utasítás

Szintaxis: WFILL x1, y1, x2, y2,[kód][, szín][, fényesség]

Értelmezés: Képernyőablak feltöltése adott kódú és/vagy adott színű karakterekkel.

x1, y1: Képernyőablak bal felső sarka.

x2, y2: Képernyőablak jobb alsó sarka.

Kód: Egy egész szám a 0-255 intervallumban, a feltöltő karakter képernyőkódja.

Szín: Egy egész szám az 1-16 intervallumbán, a feltöltő karakter színe.

Fényesség: Egy egész szám a 0-7 intervallumban, a feltöltő karakter fényessége.

Példa:

```
100 FOR I=0 TO 255
110 : X1=RND(0)*20
120 : Y1=RND(0)*15
130 : WFILL X1,Y1,X1+20,Y1+10,I
140 NEXT I
```

WIDTH utasítás

Szintaxis: WIDTH n

Értelmezés: Vonalvastagság beállítása minden grafikus utasítás számára (BOX,CIRCLE,DRAW stb.).

n: Értéke 1 vagy 2 lehet.

1: Vonalvastagság 1 képpont.
2: Vonalvastagság 2 képpont.

WINDOW utasítás

szintaxis: WINDOW CLR
WINDOW x1, y1, x2, y2[, mód]

Értelmezés: Teljes képernyő visszaállítása, képernyőablak definiálása,

törlése, képernyő-téglalap villogtatása, inverzbe forgatása a karakteres képernyőn.

x1, y1: Képernyőablak bal felső sarka.

x2, y2: Képernyőablak jobb alsó sarka.

mód: Egy egész szám a 0-3 intervallumban. Alapesetben 0.

0: Képernyőablak megnyitása és HOME
1: Képernyőablak megnyitása és CLR
2: Képernyő-téglalapban FLASH ON / FLASH OFF
3: Képernyő-téglalapban RVS ON / RVS OFF

Példa:

```
100 WINDOW 10,10,20,20,1
110 LIST
120 WINDOW 5,5,34,20,3
130 GETKEY E$
140 WINDOW CLR
150 LIST
160 WINDOW 0,0,39,24,2
```

WOPEN utasítás

Szintaxis: WOPEN x1, y1, x2, y2[, mód]

Értelmezés: Képernyő-téglalap tartalmának elmentése pufferbe, ahonnan a WCLOSE utasítással visszaállítható a WOPEN utasítás kiadása előtti állapot.

x1, y1: Képernyő-téglalap bal felső sarka.

x2, y2: Képernyő-téglalap jobb alsó sarka.

Mód: Egy egész szám a 0-3 intervallumban. Alapesetben 0.

0: Képernyő-téglalap elmentése pufferbe
1: Képernyő-téglalap elmentése pufferbe és törlése.
2: Képernyő-téglalap elmentése pufferbe és keret kiírása.
3: Képernyő-téglalap elmentése pufferbe, törlése és keret kiírása.

Megjegyzés: A WOPEN- és WCLOSE utasítások a $F000-$F7FF-ig terjedő memóriarészt használják képernyő-puffernek.

Példa:

```
100 DO WHILE RKEY(0)=0
110 : X1=RND(0)*20
120 : Y1=RND(0)*15
130 : WOPEN X1,Y1,X1+20,Y1+10,3
140 : LIST
150 LOOP
```

WORD utasítás

Szintaxis: WORD cím, érték
WORD cím, érték1[, érték2][,...][, értékn]

Értelmezés: A WORD utasítás azonos a POKE utasítással, de a tárolható érték 0-65535-ig terjedhet. A tárolás két byte-on történik, előbb az alsó, majd a felső byte kerül elhelyezésre a memóriában a megadott címtől kezdődően. A betöltendő értékből vesszővel elválasztva több is megadható, ilyenkor azok az előbb leírt módon, egymás mögé kerülnek elhelyezésre a memóriában.

Megjegyzés: WORD utasítással az alábbi müveletek helyettesíthetök.

```
100 HI=INT(ÉRTÉK/256)
110 LO=ÉRTÉK-256*HI
120 POKE CÍM,LO
130 POKE CÍM+1,HI

100 WORD CÍM,ÉRTÉK
```

Példa:

```
100 WORD 55,DEC("C000"):CLR
```

A BASIC memória végét $C000-ra állítja, helyet hagyva a sprite-definíciók számára. Bekapcsoláskor a BASIC memória vége $E000.

XOR függvény

Szintaxis: v=XOR (n1, n2)

Értelmezés: Kizáró vagy müveletet végez el két szám között.

n1, n2: Két egész szám 0-65535 intervallumban.

Példa:

```
PRINT XOR(124,12);XOR(0,12)
112 12
READY.
```

Kibővített BASIC 3.5 utasítások:

RUN utasítás

Szintaxis: RUN [sorszám]
RUN filenév[,D meghajtó][,U egységszám]

Értelmezés: Lefuttat egy a memóriában levő BASIC-programot, vagy betölt egyet lemezről és elindítja.

Sorszám: Egy érvényes programsor sorszáma, a futtatás kezdősorszáma.

Filenév: Egy kifejezés idézőjelek között vagy változóban, maximum 16 karakter hosszban.

Meghajtó: 0 vagy 1, attól függően melyik meghajtót kivánjuk használni. Alapesetben 0.

Egységszám: Egy egész szám a 4-15 intervallumban. Alapesetben 8.

Megjegyzés: A sorszámot változóban is megadhatjuk.

A filenév, meghajtó és egységszám változóban is megadható, ilyenkor a változót zárójelbe kell tenni.

Példa 1:

```
100 FOR I=1 TO 2:PRINT I;:NEXT
200 PRINT "HALLO!"
RUN
1 2HALLO!
READY.
RUN 200
HALLO!
READY.
```

Példa 2:

```
RUN"PROGR1",D0,U9
```

Betölti a PROGR1 file-t a 9-es egységszámú lemezegység 0-ás meghajtójában lévő lemezről és elindítja.

RESTORE utasítás

Szintaxis: RESTORE [sorszám]

Értelmezés: Beállítja a READ utasítás mutatóját

Sorszám: Egy érvényes programsor sorszáma.

Megjegyzés: A sorszám változóban is megadható.

Példa:

```
100 DATA 1,2,3,4,5,6
110 READ U,V,W
120 D=100
130 RESTORE D
140 READ X,Y,Z
150 PRINT U;V;W;X;Y;Z
RUN
 1 2 3 1 2 3
READY.
```

RESUME utasítás

Szintaxis: RESUME
RESUME NEXT
RESUME [sorszám]

Értelmezés: A hibakezelő rutinra ugrás esetén (TRAP utasítás) eldönti a folytatás módját.

Sorszám: Egy érvényes programsor sorszáma.

Megjegyzés: A sorszám változóban is megadható.

Példa:

```
100 TRAP 500
.
.
.
500 IF ER=11 THEN RESUME 250
```

Egy SYNTAX ERROR hibaüzenet esetén a vezérlés a 250-es sorra kerül.

ON...CALL,
ON...CGOSUB,
ON...CGOTO,
ON...JUMP,
ON...RESTORE és
ON...RESUME utasítások

Szintaxis: ON n CALL szubrutinnévlista
ON n CGOSUB sorszámlista
ON n CGOTO sorszámlista
ON n JUMP szubrutinnévlista
ON n RESTORE sorszámlista
ON n RESUME sorszámlista

Értelmezés: Elágazás az n kifejezés értéke szerint.

n: Tetszőleges aritmetikai kifejezés.

Sorszámlista: Érvényes programsor sorszámok egymástól vesszővel elválasztva.

Szubrutinnévlista: Érvényes szubrutinnevek, idézőjelek között és egymástól vesszővel elválasztva.

Megjegyzés: A sorszám és szubrutinnév változóban is megadható.

Példa 1:

```
100 ON L-1 CGOTO 150,300,320,220
```

L=2 értékénél a vezérlés a 150-es, L=3-nál a 300-as sorra stb. kerül.

Példa 2:

```
100 ON M JUMP "DRIVE/A","DRIVE/B","KUKA"
...
285 SUB"KUKA"
...
420 SUB"DRIVE/A"
...
612 SUB"DRIVE/B"
```

M=1 értékénél a vezérlés a "DRIVE/A", M=2-nél a "DRIVE/B" cimkével megjelölt programsorra stb. kerül.

GRAPHIC utasítás

Szintaxis: GRAPHIC CLR
GRAPHIC mód[,[törlés],szövegsor]

Értelmezés: Felszabadítja vagy lefoglalja a grafikus képernyőhöz szükséges memoriaterületet, és grafikus üzemmódot vált.

Mód: Egy egész szám a 0-4 intervallumban.
0: Karakteres képernyő.
1: Nagyfelbontású grafika (320x200 pont).
2: Nagyfelbontású grafika és karakteres képernyő, osztott üzemmód.
3: Többszínű grafika (160x200 pont).
4: Többszínű grafika és karakteres képernyő, osztott üzemmód.

Törlés: 1 megadása esetén törli a megadott képernyő(ke)t, 0 esetén nem.

Szövegsor: Egy egész szám a 0-24 intervallumban, 2-es és 4-es módus esetén a karakteres képernyő kezdetének helye. Alapesetben 19.

Példa:

```
100 GRAPHIC 4,1,15
```

BOX,
CIRCLE,
DRAW,
PAINT utásítások

Szintaxis:

```
BOX mintaszám,x1,y1[,[x2,y2]][,[szög]][,kitöltés]
CIRCLE mintaszám,[x,y],xr[,[yr][,[s]][,[v]][,[szög]][,szegmens]
DRAW mintaszám,[x,y1]TOx2,y2...
PAINT mintaszám[,[x,y][,mód]]
```

Értelmezés: A BOX-, CIRCLE-, DRAW- és PAINT utasítások használatánál a szindefiníció helyett a vonaltáblázatban definiált (PATTERN utasítás) vonal- vagy területminták valamelyikére kell hivatkozni. A vonaltáblázatban max. 16 mintát lehet egyszerre tárolni, ezekre a 0-15 intervallumban megadott mintaszámmal lehet hivatkozni. A kiválasztott vonal- vagy területmintákkal rajzolnak a grafikus utasítások.

Megjegyzés: FONTOS! A PAINT utasítás működése kb 1.2 kbyte memóriát igényel a BASIC munkaterületből.

A PAINT utasítás a mód paraméterben megadott színű pontokig vizsgálja a határokat.

mód: Egy egész szám a 0-3 intervallumban. Alapesetben 3.

0: Háttérszín
1: Előtérszín
2: Többszín #1
3: Többszín #2

Példa:

```
100 GRAPHIC 3,1
100 PATTERN 10,DEC("0055"),DEC("AAFF")
110 BOX 10,10,10,80,100,45,1
120 PATTERN 1
130 CIRCLE 3,140,150,40,40
140 CIRCLE 2,140,150,35,35
150 CIRCLE 1,140,150,30,30
160 PAINT 1,140,150,2
```

Új BASIC-hibaüzenetek:

Kód Jelentés

37 BEND NOT FOUND (BEGIN-utasítás BEND-utasítás nélkül).

38 LINE NUMBER TOO LARGE (Túl nagy a sorszám).

39 UNRESOLVED REFERENCE (Határozatlan hivatkozás).

40 UNIMPLEMENTED COMMAND (Az utasítás nem végrehajtható).

41 FILE READ (A file olvasásra van megnyitva).

42 UNDEF'D SUBROUTINE (Definiálatlan szubrutin).

43 END SUB WITHOUT CALL (END SUB-utasítás CALL-utasítás nélkül).

44 EXIT SUB WITHOUT CALL (EXIT SUB-utasítás CALL-utasítás nélkül).

Rendszerváltozók és I/O címek kiegészítés:

```
        $00D0-$00E8 ;nullalapos átmeneti tárolók
        $02CC-$02E3 ;nem nullalapos átmeneti tárolók
;
LPAGE   $053D ;listázás oldalhossza. Alapesetben 0.
;
LNGJMP  $05F0 ;hosszú ugrás vektor
        $05F1 ;
FETARG  $05F2 ;hosszú ugrás akku
FETXRG  $05F3 ;hosszú ugrás x regiszter
FETSRG  $05F4 ;hosszú ugrás státusz regiszter
;
        $05F5-$0608 ;Indítóprogram a 3-PLUS-1 programhoz
;
        $05F5 LDX #$05   ;lapozókód
        $05F7 LDA #$80   ;hosszú ugrás cím: $8003
        $05F9 STA $05F1  ;
        $05FC LDA #$03   ;
        $05FE STA $05F0  ;
        $0601 STA $05F4  ;hosszú ugrás státusz
        $0604 LDA $FB    ;aktuális lapkód elhozás
        $0606 JMP $FCFA  ;hosszú ugrás rutin
;
        $060A-$061B ;Indítóprogram az OCTASOFT BASIC-hez
;
        $060A LDX #$03   ;
        $060C STX $05F0  ;hosszú ugrás cím: $8003
        $060F STX $05F4  ;hosszú ugrás státusz
        $0612 LDA #$80   ;
        $0614 STA $05F1  ;
        $0617 LDA $FB    ;aktuális lapkód elhozás
        $0619 JMP $FCFA  ;hosszú ugrás rutin
;
        $061C-$0693 ;Segédrutinok a RAM vektortáblához:
;
TCINV   $061C LDX #$03     ;interrupt rutinra
        $061E STA $FDD0,X  ;
        $0621 JMP $8356    ;
;
TBRK    $0624 LDA $0106,X  ;
        $0627 BPL $0639    ;
        $0629 CMP #$C0     ;
        $062B BCS $0639    ;
        $062D LDX #$03     ;
        $062F CPX $FB      ;
        $0631 BNE $0639    ;
        $0633 STA $FDD0,X  ;
        $0636 JMP $822C    ;
        $0639 LDX #$00     ;
        $063B STX $FB      ;
        $063D STA $FDD0,X  ;
        $0640 JMP $F44C    ;monitor break
;
TERROR  $0643 PHP          ;hibaüzenet vektor
        $0644 PHA          ;
        $0645 LDA #$00     ;
        $0647 BEQ $066B    ;
;
TMAIN   $0649 PHP          ;direkt rendszerciklus
        $064A PHA          ;
        $064B LDA #$02     ;
        $064D BNE $066B    ;
```

```
TQPLOP $064F PHP          ;listázó rutin
       $0650 PHA          ;
       $0651 LDA #$04     ;
       $0653 BNE $066B    ;
;
TGONE  $0655 PHP          ;karakter átlépés
       $0656 PHA          ;
       $0657 LDA #$06     ;
       $0659 BNE $066B    ;
;
TEVAL  $065B PHP          ;szimbólum kiértekelés
       $065C PHA          ;
       $065D LDA #$08     ;
       $065F BNE $066B    ;
;
TESCLK $0661 PHP          ;parancs feldolgozás
       $0662 PHA          ;
       $0663 LDA #$0A     ;
       $0665 BNE $066B    ;
;
TLOAD  $0667 PHP          ;load vektor
       $0668 PHA          ;
       $0669 LDA #$0C     ;
;
       $066B STX $05F3    ;
       $066E LDX #$03     ;
       $0670 STX $FB      ;
       $0672 STA $FDD0,X  ;
       $0675 TAX          ;
       $0676 LDA $8330,X  ;
       $0679 STA $05F0    ;
       $067C LDA $8331,X  ;
       $067F STA $05F1    ;
       $0682 LDX #$03     ;
;
TJUMP  $0684 STX $FB      ;
       $0686 STA $FDD0,X  ;
       $0689 LDX $05F3    ;
       $068C PLA          ;
       $068D PLP          ;
       $068E JMP ($05F0)  ;
;
TUNIMP $0691 JMP $89DF;nem végrehajtható utasítás
;
       $0694-$06EB ;nem használt
;
FFRMSK $07FA ;képernyőosztás helye sorszámban (0-25). Alapesetben 19.
;
VMBMSK $07FB ;programozható karaktergenerátor helye. Alapesetben $E0.
             ;b7-b2: karaktergenerátor kezdőcíme kbyte-ban
             ;b1-b0: nem használt
;
TEDMSK $07FE ;aktuális TED üzemmód. Alapesetben $04.
             ;b7: 0: inverz megjelenítés bekapcsolva
             ;    1: inverz megjelenítés kikapcsolva
             ;b6: 0: bővített háttérszínű mód kikapcsolva
             ;    1: bővített háttérszínű mód bekapcsolva
             ;b5: nem használt
             ;b4: 0: többszínű mód kikapcsolva
             ;    1: többszínű mód bekapcsolva
             ;b3: nem használt
             ;b2: 0: karaktergenerátor a RAM-ban ($07FB általi kezdőcímen)
             ;    1: karaktergenerátor a ROM-bam ($D000 kezdőcímen)
             ;b1: nem használt
             ;b0: nem használt
```

```
PRGMSK $07FF  ;aktuális program üzemmód. Alapesetben $03.
              ;b7: OPTION flag
              ;b6: nem használt
              ;b5: RCOMP flag
              ;b4: nem használt
              ;b3: DRIVE flag
              ;     0: DRIVE OFF
              ;     1: DRIVE ON
              ;b2: nem használt
              ;b1-b0: aktuális sprite-bank sorszáma
;
TEDATR $0800-$0BFF ;szöveges képernyő attributum tárterület:
              ;b7: villogás flag
              ;     0: nem villog
              ;     1: villog
              ;b6-b4: intenzitás (0-7)
              ;b3-b0: szín (0-15)
;
TEDSCN $0C00-$0FFF ;szöveges képernyő karakterkód tárterület:
              ;b7: inverz flag
              ;     0: nem inverz
              ;     1: inverz
              ;b6-b0: karakterkód (0-127)
;
SCBUFF $1000-$13FF ;átmeneti tároló sprite-okhoz (PROJECT utasítás)
SPBUFF $1400-$15FF ;átfedés tároló sprite-okhoz (PROJECT utasítás)
SYSREG $1600-$16FF ;rendszerváltozó tárterület bővítés
PATBUF $1700-$17FF ;kitöltési minta tárolóterület (PATTERN utasítás)
;
BASMEM $1800-$DFFF ;BASIC programterület grafikus képernyő nélkül
;
BMLUM  $1800-$1BFF ;grafikus kép intenzitásinformáció terület
BMCOLR $1C00-$1FFF ;grafikus kép színinformáció terület
GRBASE $2000-$3FFF ;grafikus kép bitpont képinformáció tárterület
BASMEM $4000-$DFFF ;BASIC programterület grafikus képernyővel
;
CHSET1 $E000-$E7FF ;1. programozható karaktergenerátor
CHSET2 $E800-$EFFF ;2. programozható karaktergenerátor
WBUFF  $F000-$F7FF ;képernyő tárolóterület (WOPEN, WCLOSE utasítások)
SEBUFF $F800-$FCFF ;átmeneti tároló sprite editáláshoz (SPRDEF utasítás)
;
       $FD00-$FF3F I/O terület
;
       $FF40-$FFFF nem használt
;
;A $1600-$16FF rendszerváltozó tárterület részletezése:
;
MOBX    $1600 ;sprite x pozició (b0-b7) (MOVSPR utasítás)
MOBY    $1608 ;sprite y pozició (b0-b7) (MOVSPR utasítás)
MOBXH   $1610 ;sprite x pozició legfelső bit (b8) (MOVSPR utasítás)
MOBENA  $1611 ;sprite engedélyezés (SPRITE utasítás)
MOBPRI  $1612 ;sprite-háttér prioritás (SPRITE utasítás)
MOBMUL  $1613 ;sprite többszínű mód (SPRITE utasítás)
MOBMOS  $1614 ;sprite-sprite ütközés regiszter
MOBDAT  $1615 ;sprite-háttér ütközés regiszter
MOBCOL  $1616 ;sprite szín (SPRCOLOR utasítás)
MOBDEF  $161E ;sprite definíció (SPRITE utasítás)
MOBUP   $1626 ;sprite-ot felfelé mozgató billentyű kódja (SPEED utasítás)
MOBDN   $162E ;sprite-ot lefelé mozgató billentyű kódja (SPEED utasítás)
MOBLF   $1636 ;sprite-ot balra mozgató billentyű kódja (SPEED utasítás)
MOBRT   $163E ;sprite-ot jobbra mozgató billentyű kódja (SPEED utasítás)
MOBSPX  $1646 ;sprite x irányú maximális sebesség (DRIVE utasítás)
MOBSPY  $164E ;sprite y irányú maximális sebesség (DRIVE utasítás)
MOBTI1  $1656 ;sprite gyorsulási idő (DRIVE utasítás)
```

```
MOBTI2 $165E ;sprite lassulási idő (DRIVE utasítás)
MOBMVX $1666 ;sprite x irányú sebesség összetevő (MOVSPR utasítás)
MOBMVY $166E ;sprite y irányú sebesség összetevő (MOVSPR utasítás)
MOBSAX $1676 ;sprite x irányú pillanatnyi sebesség
MOBSAY $167E ;sprite y irányú pillanatnyi sebesség
MOBTA1 $1686 ;sprite pillanatnyi gyorsulási idő
MOBTA2 $168E ;sprite pillanatnyi lassulási idő
MOBNUM $1696 ;aktuális sprite sorszám
;
LINBUF $1697 ;grafikus vonaldefiníció hossz táblázat (PATTERN utasítás)
;
ADBUFF $16A7 ;burkológörbe felfutási/lefutási idő (ENVELOPE utasítás)
SRBUFF $16AF ;burkológörbe kitartási/lecsengési idő (ENVELOPE utasítás)
SWBUFF $16B7 ;burkológörbe kitartási szint/hullámforma (ENVELOPE utasítás)
VOLUME $16BF ;beállított hangerő (VOL utasítás)
TRATE  $16C0 ;beállított lejátszási sebesség komplemense (TEMPO utasítás)
OCTAVE $16C1 ;beállított oktáv (PLAY utasítás)
STIME  $16C2 ;beállított hanghosszúság (PLAY utasítás)
TIME   $16C3 ;segédregiszter (PLAY utasítás)
WXMIN  $16C4 ;képernyő-téglalap bal felső sarok x pozició (WOPEN utasítás)
WYMIN  $16C5 ;képernyő-téglalap bal felső sarok y pozició (WOPEN utasítás)
WXMAX  $16C6 ;képernyő-téglalap jobb alsó sarok x pozició (WOPEN utasítás)
WYMAX  $16C7 ;képernyő-téglalap jobb alsó sarok y pozició (WOPEN utasítás)
FRADEF $16C8 ;ablakkeret karakterdefiníció képernyőkódok (FRAME utasítás)
PATNUM $16D1 ;aktuális grafikus bitminta sorszáma (PATTERN utasítás)
;
       $16D2-$16DB ;nem használt
;      $16DC-$16FF ;sprite-regiszter tároló
```

OCTASOFT BASIC V7.0 tokenek:

Minden Commodore BASIC nyelv tokenek rendszerét használja a BASIC kulcsszavak (parancsok, utasítások és függvények) tárolására. Egy token a programban, a szó helyén tárolt kódot jelenti. Például a RESTORE kulcsszó hét karaktere helyett egy $8C (140) kód található. A token rendszer használata rövidebb programot és gyorsabb végrehajtást eredményez.
A C-PLUS/4 BASIC egy byte-os tokeneket használ. Egy byte-on $00-$FF (0-255) érték tárolható, ezek közül a 128-nál nagyobb kódok jelentenek kulcsszavakat, tehát összesen 128 különböző token adható meg.
Az OCTASOFT BASIC-nek 248 kulcsszava van, több mint az előbb meghatározott maximális 128, tehát a tárolásra más módot kellett keresni.
Kézenfekvő megoldást a két byte-os tokenek használata. Az OCTASOFT BASIC új-utasításait két byte-os kódok jelentik. Az új függvények első kódja egy $CE (206) byte, a második pedig egy $02-$14 közöttibyte. Az új utasításokat, parancsokat egy $FE (254) és egy azt követő $02-$5A közötti kód jelenti.
Az OCTASOFT BASIC bővítés tokenjeinek teljes felsorolását tartalmazza a következő táblázat. A '*'-gal megjelölt kulcsszavak a C-PLUS/4 hardver lehetőségének szűkössége miatt nincsenek kidolgozva!

Kód:	Kulcsszó:	Kód:	Kulcsszó:	Kód:	Kulcsszó:	Kód:	Kulcsszó:
80	END	A0	CLOSE	C0	TAN	E0	CHAR
81	FOR	A1	GET	C1	ATN	E1	BOX
82	NEXT	A2	NEW	C2	PEEK	E2	CIRCLE
83	DATA	A3	TAB(	C3	LEN	E3	GSHAPE
84	INPUT#	A4	TO	C4	STR$	E4	SSHAPE
85	INPUT	A5	FN	C5	VAL	E5	DRAW
86	DIM	A6	SPC(	C6	ASC	E6	LOCATE
87	READ	A7	THEN	C7	CHR$	E7	COLOR
88	LET	A8	NOT	C8	LEFT$	E8	SCNCLR
89	GOTO	A9	STEP	C9	RIGHT$	E9	SCALE
8A	RUN	AA	USING	CA	MID$	EA	HELP
8B	IF	AB	-	CB	GO	EB	DO
8C	RESTORE	AC	*	CC	RGR	EC	LOOP
8D	GOSUB	AD	/	CD	RCLR	ED	EXIT
8E	RETURN	AE	↑	CE	RLUM	EE	DIRECTORY
8F	REM	AF	AND	CF	JOY	EF	DSAVE
90	STOP	B0	OR	D0	RDOT	F0	DLOAD
91	ON	B1	>	D1	DEC	F1	HEADER
92	WAIT	B2	=	D2	HEX$	F2	SCRATCH
93	LOAD	B3	<	D3	ERR$	F3	COLLECT
94	SAVE	B4	SGN	D4	INSTR	F4	COPY
95	VERIFY	B5	INT	D5	ELSE	F5	RENAME
96	DEF	B6	ABS	D6	RESUME	F6	BACKUP
97	POKE	B7	USR	D7	TRAP	F7	DELETE
98	PRINT#	B8	FRE	D8	TRON	F8	RENUMBER
99	PRINT	B9	POS	D9	TROFF	F9	KEY
9A	CONT	BA	SQR	DA	SOUND	FA	MONITOR
9B	LIST	BB	RND	DB	VOL	FB	-
9C	CLR	BC	LOG	DC	AUTO	FC	UNTIL
9D	CMD	BD	EXP	DD	PUDEF	FD	WHILE
9E	SYS	BE	COS	DE	GRAPHIC	FE	-
9F	OPEN	BF	SIN	DF	PAINT	FF	π

Kód:	Kulcsszó:	Kód:	Kulcsszó:	Kód:	Kulcsszó:	Kód:	Kulcsszó:
FE00	-	FE20	-	FE40	PUFF$	CE00	-
FE01	-	FE21	FETCH	FE41	PUFF	CE01	-
FE02*	BANK	FE22	-	FE42*	UNDEF	CE02*	POT
FE03*	FILTER	FE23	SWAP	FE43	PRON	CE03	BUMP
FE04	PLAY	FE24*	OFF	FE44	PROFF	CE04*	PEN
FE05	TEMPO	FE25	FAST	FE45	DISAPA	CE05	RSPPOS
FE06	MOVSPR	FE26	SLOW	FE46	SECURE	CE06	RSPRITE
FE07	SPRITE	FE27	FILL	FE47	NORMAL	CE07	RSPCOLOR
FE08	SPRCOLOR	FE28	REFRESH	FE48	EXTEND	CE08	XOR
FE09	RREG	FE29	PROJECT	FE49	MULTI	CE09	RWINDOW
FE0A	ENVELOPE	FE2A	BVERIFY	FE4A	PATTERN	CE0A	POINTER
FE0B	SLEEP	FE2B	HARDCOPY	FE4B	DESK	CE0B	RSPLUM
FE0C	CATALOG	FE2C	DEFCHR	FE4C	SIZE	CE0C	RCBANK
FE0D	DOPEN	FE2D	CENTRE	FE4D	RESET	CE0D	RSBANK
FE0E	APPEND	FE2E	SUB	FE4E	ONKEY	CE0E	PEEK$
FE0F	DCLOSE	FE2F	CALL	FE4F	GETIN	CE0F*	MOD
FE10	BSAVE	FE30	JUMP	FE50	CBANK	CE10*	DIV
FE11	BLOAD	FE31	OLD	FE51	SBANK	CE11*	FRAC
FE12	RECORD	FE32	FIND	FE52	DRIVE	CE12*	STRING$
FE13	CONCAT	FE33	CHANGE	FE53	SPEED	CE13	INKEY
FE14	DVERIFY	FE34*	DUMP	FE54	CGOTO	CE14	RKEY
FE15	DCLEAR	FE35*	SLIST	FE55	CGOSUB	CE15	-
FE16	SPRSAV	FE36	PAGE	FE56	WOPEN	CE16	-
FE17	COLLISION	FE37	OPTION	FE57	WCLOSE	CE17	-
FE18	BEGIN	FE38	SCROLL	FE58	WFILL	CE18	-
FE19	BEND	FE39	MOVE	FE59	FRAME	CE19	-
FE1A	WINDOW	FE3A	-	FE5A	INBOX	CE1A	-
FE1B	BOOT	FE3B	RCOMP	FE5B	-	CE1B	-
FE1C	WIDTH	FE3C	ESC	FE5C	-	CE1C	-
FE1D	SPRDEF	FE3D	MERGE	FE5D	-	CE1D	-
FE1E	QUIT	FE3E	WORD	FE5E	-	CE1E	-
FE1F	STASH	FE3F*	FLOAD	FE5F	-	CE1F	-

TARTALOMJEGYZÉK